REMARKS

In view of the foregoing amendments and following remarks responsive to the Office Action dated October 4, 2005, Applicant respectfully request's favorable reconsideration of this application.

Rejections and Objections as to Form

In section 1 of the Office Action, the Office objected to the disclosure as containing an embedded hyperlink. Applicant has herein amended the specification so that the hyperlink is now not browser-executable.

In section 2 of the Office Action, the Office noted that all trademarks used in the application should be capitalized and accompanied by the generic terminology. Applicant has herein amended the specification accordingly.

In sections 3 and 4 the Office Action, the Office rejected claims 1-25 under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. Specifically, the Office asserted that the preamble of these claims defines the apparatus as a computer program product recorded on a computer readable medium and that the end of the claim recites that the computer program product is in the form of a Web service. The Office asserted that it is unclear how a computer program product recorded upon a computer readable medium is a Web service since a Web service is a method that can be performed over the World Wide Web.

While Applicant is not certain that it understands the nature of this rejection, as part of its overall strategy, Applicant has amended the claims to eliminate this use of language. Accordingly, this rejection is now moot.

In section 5 of the Office Action, the Office again rejected claims 1-25 as not complying with the enablement requirement because the second clause of claim 1 recites that the container transmits to other containers and it is unclear how a computer program product embodied a computer readable media transmits to another computer program product on computer readable media.

Applicant has herein amended claim 1 so as not to directly recite the transmitting aspect, which, technically, might be deemed to be performed by the network or, at least, other equipment or software on the network, rather than the container itself. This amendment should eliminate the grounds for this rejection.

In sections 6-18 of the Office Action, the Office rejected all claims, claims 1-48, under 35 U.S.C. 112, second paragraph, as being indefinite listing in detail specific rejections. Applicant has herein generally amended the claims in order to improve their form. Thus, many of the issues raised in the 112 rejections are now moot in view of the new claim language. In any event, Applicant addresses these rejections individually below with reference to the claims as newly amended.

In section 8, the Office asserted that claim 25 recites the limitation "said computer program" without sufficient antecedent basis. Applicant has herein amended claim 25 to correct this oversight.

In section 9 of the Office Action, the Office rejected claims 1-4, 7, 11-19, 21-24, and 26 for referring to "computer readable code". The Office asserted that it is unclear what Applicant means by "computer readable code" or how "computer readable code" would be embodied in order to make said code computer readable.

Applicant does not understand this rejection. There can be little doubt that a person of ordinary skill in the art of computer programming understands what computer readable code is and how to embody it in a medium so that it is computer readable. Nevertheless, Applicant has amended the language of the claims so that it no longer uses this terminology. The claims now refer to the code as computer executable instructions, which is well-accepted terminology in the USPTO for claiming software.

In section 10 of the Office Action, the Office rejected all claims asserting that it is unclear where the messages being transmitted to other containers are being originated. The Office asserted that the claims and specification are unclear as to whether the originator is the node which will receive the services, a program separate but local to the node, or an outside originating entity.

Applicant respectfully traverses. The preamble of claim 1, for instance, clearly recites that the "computer program product" being claimed is called a "container". The second paragraph of claim 1 recites that the "container" comprises computer code for generating messages to be transmitted to other containers. Claim 1 further recites that the container "corresponds to" a particular network node. For instance, see elements 1, 2, 3, and 4 of claim 1, all of which refer to containers corresponding to particular nodes. As a practical matter, this typically will mean that the container is local to that node. However, Applicant has intentionally written the claim using the "corresponding to" language so as not to be so limited. Thus, it is clear from the language of claim 1 that (1) each "container" is a program that corresponds to a node and (2) that the recited code for generating messages is part of the container recited in the preamble. Thus, the claim clearly recites that the messages are originated at the container, which is a program corresponding to a particular node. Thus, the claim language is clear. Hence, in direct response to the Office's inquiry as to whether the originator is the node which will receive the services, a program separate but local to the node, or an outside originating entity; the originator is a program corresponding to the node (of which a program "local to the node" would be the most practical embodiment).

In view of the foregoing, Applicant respectfully requests the Office to withdraw this rejection in view of the currently pending claims.

With respect to independent method claim 25, it has very similar language to the language of claim 1 discussed above. Hence, claim 25 is sufficiently definite for the same reasons discussed with respect to claim 1.

In section 11 of the Office Action, the Office rejected claims 1-48 as unclear as to what Applicant means by "dynamically reconfiguring Web services based on said messages and said Web services available at said corresponding network node". The Office explained that the language is confusing as it is unclear if the messages are at the corresponding network node, the Web services are active at the corresponding network node, the Web services can be downloaded at the corresponding network node, or if the messages are local.

The amendments to independent claims 1 and 25 cause these claims to more clearly recite what messages are being referred to and where they originate. Applicant has rewritten this element of the claims to more precisely recite the invention. As noted in the specification at, for instance, page 6, first sentence, and the numerous reconfiguration examples provided on pages 19-39, the invention provides a software construct, herein termed a Web service "container", for managing Web services at a network node and an adaptive model for the dynamic configuration of a plurality of Web service containers distributed throughout a network. Thus, the Web services that are configured in accordance with the principles of the present invention are essentially all of the Web services on the network. This is accomplished using the messages sent back and forth across the network by the various containers communicating with each other. Applicant has amended the language of claims 1 and 25 to more precisely recite this concept.

In section 12 of the Office Action, the Office rejected claim 2 asserting that it is unclear what the difference is between requesting a copy of Web services software and receiving requested Web services software. The Office asserted that the claim language is unclear as to whether the requested Web services software being received is in fact a copy of the Web services software being requested.

Applicant has herein amended claim 2 to make the language between the two clauses more consistent, thereby eliminating the grounds for this rejection.

In section 13 of the Office Action, the Office rejected claim 3 asserting that the wording confuses the transmission, receipt, and generation of messages. Particularly, the Office asserted that it is confusing if the messages being generated are, in fact, the messages being transmitted and received.

Applicant has herein amended claim 3 in order to more clearly recite that there are transmitted messages and, separately, there are received messages.

In section 14 of the Office Action, the Office rejected claim 4 asserting that it is confusing whether the Web services registry is a container. Applicant has amended claim 4 by, *inter alia*, adding the word "further". This should clarify that the Web services registry is not a container[1].

In section 15 of the Office Action, the Office further rejected claims 7 and 12 asserting that it is confusing whether the messages disclosing Web services available at network nodes detail if these services are available locally or on other network nodes beside the local requester.

[1] Of course, the claim does not recite that the Web services registry is not a container. In typical drafting fashion, the claim simply does not recite whether or not it is a container.

Claim 7 refers back to the messages referred to in clauses 2 and 3 of claim 1. Applicant has herein amended claim 7 so as to greatly simplify the language while still clearly referring back to the messages previously recited in clauses 2 and 3 of claim 1. Accordingly, there should be no confusion as to what messages are being discussed in these claims. The amendments to claim 1 discussed above in connection with section 10 of the Office Action (concerning where the messages being transmitted to other containers are being originated) also address the issue raised here in section 15 of the Office Action.

With respect to claim 12, Applicant has made minor amendments to claim 12 in order to more clearly recite that the nodes are other network nodes.

In section 16 of the Office Action, the Office rejected claim 13 asserting that it is unclear whether the response being received to said request is a message indicating the Web service can be downloaded or is the downloaded Web service software.

Applicant respectfully traverses as it believes that claim 13 is clear. The claim recites "computer executable instructions for returning said responses to said requesting clients". There is only one recitation of "responses" in claim 13 or any claim from which it depends, namely, the responses to said requests. Accordingly, there should be no question, that claim 13 recites that the responses are the messages indicating that the Web services are available, and not the Web services themselves.

Therefore, Applicant kindly requests the Office to withdraw this rejection.

In section 17 of the Office Action, the Office rejected claim 14 asserting that it is unclear based upon claim dependency whether the request is actually initially issued by a "container".

Applicant has herein amended claim 14 to more clearly recite that the request referred to is a request routed from another container as recited in claim 13. Thus, the request initially is issued by a client machine to another container and that other container simply routes the request to the container being discussed in claim 14. In short, the request is not initially issued by a container, but only routed from a client through a container to another container.

In section 18 of the Office Action, the Office rejected claim 17 asserting that it is unclear what Applicant means by "offloading said local code for said particular Web service". The Office asserted that it is unclear if the code is being offloaded, if the Web service is being offloaded, or if the code is being offloaded as part of the Web service.

This rejection is confusing. It is not clear what the Office perceives as the differences between the three options specified in the rejection. Nevertheless, Applicant has herein amended claim 17 to clean up the language and more clearly recite that the Web service received from the other container is offloaded when the load of client requests drops below the second predetermined level. For further clarification, see the specification at page 27, lines 1-4.

Prior Art Rejections

In section 20 of the Office Action, the Office rejected claims 1- 4, 7-9, 11-14, 22-23, 25-28, 31-33, 35-38, and 46-47 under 35 U.S.C. 102(e) as anticipated by Zintel. The Office rejected the remaining claims as obvious over Zintel in combination with various secondary references.

Applicant respectfully traverses all of the prior art rejections insofar as Zintel does not teach that for which it has been cited.

The Present Invention

The present invention relates to the maintenance and movement of Web services in a network. As described on page 3, lines 11 et seq. of the present specification, “Web services” are application logic or software modules that can be exposed to and shared with other nodes over the Internet via a standardized interface mechanism.

The invention provides a software construct (termed a Web service “container” in the specification) for managing Web services at a network node and an adaptive model for the dynamic configuration for a plurality of Web service containers distributed throughout a network in a software and hardware platform-independent manner. The containers dynamically adapt themselves and, particularly, the Web services contained therein based on a pluggable set of heuristics. The containers can exchange Web services software on a peer-to-peer basis or through querying a registry. The containers can discover the Web services contained in other containers. The present invention allows containers to dynamically exchange Web services software as well as contextual information, such as current workload, so that containers are virtually limitlessly reconfigurable based on context. For instance, containers can not only reconfigure themselves as routers to other containers containing requested Web services, but can load and unload Web service software modules based on detected workload and/or Web service availability at other servers and send software modules to peer containers as needed to allow them to run a Web service locally rather than from a remote location on the network.

The invention also enables one server with a Web services container to send Web services software to another server with a Web services container in order to allow that other server to begin providing that service. This may be useful, for instance, when the work load at a first server exceeds that server's capabilities. That server can then send the Web service software to one or more other servers and then divide the servicing of requests for that service among two or more servers. The first server can reconfigure itself either partially or totally as a "service router" to route requests for given Web services to other servers that it has determined can provide that service either by virtue of it having itself sent the Web service software to the other server(s) or by querying the other server(s) as to the contents of their Web service containers.

Another exemplary routing scenario involves a request being received by the first server. Acting as a "service router", the first server directs the request to the second server. When the second server responds to the request, it includes context information that indicates it was the node that ultimately handled the request. The container that initiated the request understands the contextual information returned and, for each subsequent request, directs the outgoing message to the second server. This can be logically thought of as dynamically adding a WSDL port to the service and indicating that the new port is the preferred endpoint. In this manner, any number of "hops" may be taken before reaching the ultimate destination, but network efficiency is gained by providing a mechanism that avoids unnecessary processing from the second service request forward.

Not only can Web services software be exchanged in a platform independent manner, but the Web service containers themselves are platform independent. That is, the Web service containers at two different network nodes can be implemented in different programming languages and run on different platforms, while still being able to exchange contextual information and Web service software modules using SOAP and WSDL.

The Zintel Reference

The Office's primary reference, Zintel, essentially has nothing to do with Web services at all or with the exchange of any form of software modules between nodes of a network. Zintel discloses a system and protocol for new nodes entering a network to register themselves on the network and make themselves known to other nodes on the network. This is a completely different issue than the maintenance, exchange, and discovery of Web service modules amongst the nodes of a network.

Discussion

As noted above, Applicant has defined what it means by "Web service" in its specification. Accordingly, the Office has either misread the Zintel reference or is defining Web services overly broadly to cover any software used in connection with a network. Either way, the prior art rejections are improper.

For exemplary purposes, let us look at the Office's anticipation rejection in connection with independent claim 1. In section 21 of the Office Action, the Office asserted that Zintel discloses determining and describing Web services that are available at a corresponding network node and refers to the simple service discovery protocol disclosed in column 12, lines 31-59. This section of Zintel has nothing to do with Web services. The Simple Service Discovery Protocol (SSDP) "is a protocol that enables Devices to learn of the existence of potential peer Devices and the required information (an IP address) needed to establish TCP/IP connections to them. The successful result of an SSDP search is a Uniform Resource Locator (URL)." Thus, unless the Office is interpreting "Web services" as any software on a network node, column 12 of Zintel has nothing to do with this element of claim 1.

Next, the Office asserted that Zintel teaches transmitting to other containers messages via a network disclosing Web services that are available at said corresponding node at column 17, lines 51-55, where it discusses that changes in an SST (Service State Table) are announced to all interested user control points (nodes). Again, this relates to identifying the properties of the nodes added to networks and does not relate to "Web services". It is instructive to review the terminology definitions set forth in Zintel at column 6, line 54 through column 12, line 20. “Services” is defined in column 9, lines 1-12 as follows:

> Service. The fundamental UPnP control entity (but not the finest level of control). An example of a Service is "Clock". Services are defined with a mandatory common base set of functionality. Vendors can extend base functionality with proprietary extensions provided the base functionality is implemented. Service definitions are versioned and later versions are constrained to be supersets of previous versions. UPnP enables searches for all Devices that contain a specified Service of a minimum version. This search would find all clocks, regardless of their packaging. The search for Device Type "Clock" would be used to find only stand-alone clocks.

Accordingly, it is quite clear that Zintel's "Services" have nothing to do with the present invention’s "Web services". Furthermore, Zintel's Service State Table discussed in column 17 and the Simple Service Discovery Protocol discussed in column 12 are essentially irrelevant to the present invention. The exchange of service information between nodes of Zintel, therefore, has nothing to do with the transmitting and receiving of messages disclosing available Web service, as recited in the second and third paragraphs of claim 1.

Next, the Office asserted that the act of "dynamically reconfiguring Web services based on said messages and said Web services available at said corresponding network node" recited in claim 1 reads on the control using UPnP networking and the Simple Service Discovery Protocol disclosed in column 57, lines 14 through column 58, line 34 of Zintel.

As previously noted, Zintel concerns the exchange of information between nodes on the network about the functionality of the nodes of the network and does not relate to discovery of "Web services" or the exchange or sharing of Web services over a network. Accordingly, the description in Zintel columns 57-58 relating to (1) discovery of devices added to a network, (2) describing new devices, (3) learning how to and then actually controlling new devices, (4) eventing (which relates to updating information about nodes by sending event messages), and (5) presenting (in which a control point can retrieve the page and, depending on the capabilities of the page, will allow a user to control the device and/or devices) has nothing to do with Web services.

However, even further, even if we were to accept the Office's expansive definition of "Web services" as encompassing any software on a network, this section of Zintel still would not read on this claim limitation. The description in columns 57 and 58 of Zintel still only relates to exchanging information about Services available at nodes and using those services. There is no discussion whatsoever of configuring the services based on any of the messages exchanged between the nodes.

Finally, the Office asserted that the limitation of claim 1 that the container is in the form of a Web service is found in Zintel because the UPnP discovery protocol occurs over the Web.

Once again, this seems to be the clearest statement that the Office is reading the term "Web services" to cover any software at a node of a network. However, as previously noted, this definition of "Web services" is inconsistent with the Applicant's express definition of "Web services" contained in its specification and simply is not a fair reading of the claim.

Hence, independent claim 1 patentably distinguishes over Zintel.

Independent claim 25 is a method claim that closely parallels the language of claim 1. Accordingly, its analysis would be very similar to that of claim 1 and, therefore, does not require repetition.

All of the dependent claims depend from one or the other of claims 1 and 25 and, therefore, distinguish over the prior art of record for at least all of the same reasons as the independent claims. None of the secondary references cited in the obviousness rejections provides the teachings discussed above that are missing from Zintel.

Nevertheless, the dependent claims add even further distinguishing features over the prior art of record. For instance, claims 2 and 26 recite the feature of requesting and receiving copies of Web services software from other nodes. The Office asserts that this is taught in column 58, lines 41-65 of Zintel, where Zintel sends messages requesting the UPnP description of a device's services and their parameters and controls and this is returned to the requester. First of all, of course, as previously noted numerous times, this has nothing to do with Applicant's Web services. However, even using the Office's expansive definition of Web services as including any software module, it still would not read on Zintel. There is a considerable difference between exchanging information about software modules and exchanging the software modules themselves. Zintel only discloses exchanging information about his "Services".

Dependent claims 4 and 28 recite the feature whereby the containers perform their discovery by querying a Web services registry. The Office asserted that Zintel's contacting of the UPnP template to find out the UPnP description for a device as discussed in column 58, lines 35-65 meets this limitation. Zintel's UPnP templates are associated with the particular node to which they correspond. In fact, as near as Applicant understands the prior art rejections in this case, the UPnP template appears to be what the Office considers (erroneously) to correspond to the claimed "containers". Thus, the Office is recharacterizing the very same template that it needs to rely on as constituting the "container" as a Web services registry. This is especially improper here where Applicant has bent over backwards to make clear that this claim recites an alternative to discovery via querying containers. Specifically, the specification discusses four different ways to discover Web services information, including(1) peer-to-peer inquiry, i.e., two containers directly communicating with each other, and (2) querying of a Web services registry. See page 11, line 8 through page 12, line 3. Thus, Applicant's specification clearly describes that the claimed Web services registry is not the claimed container. Therefore, it is improper for the Office to read Zintel's template on both the claimed "containers" and the claimed "registry", when it is so clear from Applicant's specification that they are two different things.

Even further, Web services registries are well known in the field and no one skilled in these arts would consider Zintel's template to be a Web services registry. If this is not enough, Applicant has clearly set forth what it means by "Web service registry" by referring to specific examples of available Web services directories. See page 4, lines 4-11 of the specification, where it states:

> The UDDI initiative is an XML-based registry standard by which businesses list themselves and the Web services they offer on the Internet. WSDL is one approach to describing such Web services. A key goal of the UDDI initiative is to enable companies to find each other and each other's Web services on the Internet and to make their computer systems inter-operable with each other in order to facilitate electronic commerce. The UDDI initiative allows businesses to list information about themselves, e.g., name, location, and/or the Web services they offer.

There is no rational interpretation of the term "Web services registry" that covers Zintel's template, especially when the Office already is using the template as the very "container" that this claim language is intended to distinguish from. Thus, this rejection is improper and should be withdrawn.

Dependent claims 5 and 29 recite that the messages are in WSDL. The Office asserted that this is taught in secondary reference Christensen and that it is obvious to combine Christensen with Zintel because Zintel uses XML to describe network services and Christensen teaches a format that is a known XML format variant for describing network services.

This is an improper combination of references. WSDL is an acronym for Web Services Descriptor Language. WSDL is a language developed for use in connection with Web services. The concept of using WSDL in connection with Zintel's registering of nodes on the network, which has nothing to do with Web services makes no sense. WSDL is designed for an entirely different purpose.

Furthermore, the Office rejected dependent claims 6 and 30 as being unpatentable over Zintel, Christensen and Januszewski, noting that Januszewski discusses UDDI. However, once again, the combination is improper and this rejection further highlights the fact that the Office is irrationally over broadly interpreting Applicant's claim language. First of all, Januszewski is an unnecessary reference since Applicant admits that UDDI is a known Web services registry and the Office has cited Januszewski solely to show that UDDI is a known Web services registry. Nevertheless, the proposed combination is improper because, in this combination, the Office is saying that Zintel's template can be the UDDI registry. Specifically, in rejecting claim 5 earlier, the Office asserted that Zintel's template is a Web services registry. Claims 6 and 30 recite that the claimed Web services registry is UDDI. Thus, the Office is asserting that it is obvious to make Zintel's template the well-known UDDI registry. This is inherently nonsense and this rejection should be withdrawn.

Dependent claims 11 and 35 add limitations concerning receiving requests for Web services from clients. The Office asserts that Zintel discloses this in column 57, lines 12-51, where the information about devices and services is shared over the network via messages. However, once again the Office is confusing requests for information about software modules with requests for the software modules themselves. As previously noted, these are two different things that cannot be considered equivalents of each other.

The rejections of many more dependent claims suffer from the same types of problems discussed above and should be readily apparent in view of the foregoing discussion.

Conclusion

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice of Allowance at the earliest possible date. The Examiner is invited to contact Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Dated: February 3, 2006

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

TXN:pmf

M:\TNaccarella\CLIENTS\IBM\25391\PTO\First Amendment(2).doc